CLAIMS

What is claimed is:

1. A microprocessor including:

an execution unit to trigger a snoop if a store into a memory occurs; and

a translation lookaside buffer (TLB) having a content addressable memory (CAM) and receiving a physical address indicating the location where the store occurred in the memory, the TLB to store page translations between a linear page address and a physical page address pointing to a page of the memory having contents stored within a cache, the TLB including a CAM input port responsive to the snoop to compare the physical address received by the TLB with the physical page address stored therein.

2. The microprocessor of claim 1 wherein if it is determined that the physical address received by the TLB matches a physical page address stored within the TLB, indicating that information was modified within the memory correlating to information potentially located within the cache, information within the cache is invalidated.

3. The microprocessor of claim 2 wherein information within the cache is invalidated by setting a bit in the cache to indicate invalid information in a cache line and disregarding the information within the cache.

4. The microprocessor of claim 1 wherein if it is determined that the physical address received by the TLB matches a physical page address stored within the TLB, indicating that information was modified within the memory correlating to information potentially located within the cache or a pipeline, and the microprocessor provides inclusion for the cache and the pipeline such that information within the cache and the pipeline are invalidated.

5. The microprocessor of claim 4 wherein information within the cache and a pipeline are invalidated by setting a bit in the cache to indicate invalid information in a cache line and disregarding the information within the cache and the pipeline.

6. The microprocessor of claim 4 wherein the TLB maintains original page translations for all bytes of information within the cache and pipeline to provide inclusion.

7. The microprocessor of claim 1 wherein the TLB includes data bits providing an indication as to what portion of a page in memory has its contents stored in the cache, and the microprocessor further includes snoop logic coupled to the TLB, the snoop logic to provide further comparison of the data bits with the physical address received by the TLB after a store into memory, the comparison provided by the snoop logic indicating if information was modified within a portion of a page of memory correlating to information potentially located within the cache.

8. The microprocessor of claim 7 wherein the data bits providing an indication as to what portion of a page in memory has its contents stored in the cache are the FINE HIT bits.

9. The microprocessor of claim 7 wherein if it is determined that the physical address received by the TLB matches a physical page address stored within the TLB and the

snoop logic indicates that the physical address matches the portion of memory associated with the data bits, indicating that information was modified within a portion of memory correlating to information located within the cache, information within the cache and a pipeline are invalidated.

10. The microprocessor of claim 9 wherein information within the cache and a pipeline are invalidated by setting a bit in the cache to indicate invalid information and disregarding the information within the cache and the pipeline.

11. The microprocessor of claim 9 wherein the data bits providing an indication as to what portion of a page in memory has its contents stored in the cache are the FINE HIT bits.

12. The microprocessor of claim 1 wherein the cache is a trace instruction cache and the information stored therein is instructions and the TLB is an instruction translation lookaside buffer (ITLB).

13. A method of self modifying code detection for cache coherency, comprising:

storing page table translations, the stored page table translations including linear page addresses associated with physical page addresses into a physically addressable memory for information stored into a cache memory;

providing a physical memory address of a store into the physically addressable memory; and

comparing the provided physical memory address to the physical page memory addresses included in the stored page table translations to determine if the physically addressable memory has been updated by self modifying code.

14. The method of claim 13 wherein the comparing generates a match between the provided physical memory address and one or more of the physical page memory addresses included in the stored page table translations indicating the potential occurrence of self modifying code and cache incoherency.

15. The method of claim 14 further comprising:

invalidating the instructions within the cache memory and an instruction pipeline for execution and fetching new instructions from the physically addressable memory to overwrite the invalidated instructions after the comparing generates a match indicating the potential occurrence of self modifying code and cache incoherency.

16. The method of claim 13 further comprising:

setting data bits associated with each stored page table translation indicating what portion of a page in physically addressable memory has its information stored into the cache memory.

17. The method of claim 16 further comprising:

comparing the provided physical memory address to the data bits associated with the stored page table translations to determine if the physically addressable memory has been updated by self modifying code within portions of pages in physically addressable memory.

18. The method of claim 13 further comprising:

maintaining original stored page table translations for all bytes of information within the cache memory and an instruction pipeline..

19. A computer including:

a memory; and

at least one microprocessor, the at least one microprocessor including, an instruction cache to store instructions for execution,

an execution unit coupled to the instruction cache to execute the instructions stored therein, the execution unit to trigger a snoop if a store into the memory is executed, and

an instruction translation lookaside buffer (ITLB) having a content addressable memory (CAM) and receiving a physical address indicating the location where the execution of the store occurred in the memory, the ITLB to store page translations between a linear page address and a physical page address pointing to a page of the memory having contents stored within the instruction cache, the ITLB including a CAM input port responsive to the snoop to compare the physical address received by the TLB with the physical page address stored therein.

20. The computer of claim 19 wherein if it is determined that the physical address received by the TLB matches a physical page address stored within the ITLB, indicating that an instruction was modified within the memory correlating to an instruction located within the instruction cache, instructions within the instruction cache and an instruction pipeline within the execution unit are invalidated.

21. The computer of claim 20 wherein the ITLB includes data bits providing an indication as to what portion of a page in memory has its contents stored in the instruction cache, and the microprocessor further includes snoop logic coupled to the ITLB, the snoop logic to provide further comparison of the data bits with the physical address received by the TLB after a store into the memory, the comparison provided by the snoop logic indicating if an instruction was modified within a portion of a page of the memory correlating to an instruction located within the instruction cache.

22. The computer of claim 19 wherein the instruction cache is a trace instruction cache.

23. A method of detecting cache incoherency in a computer comprising:

providing a physical address associated with a store into memory;

comparing the physical address associated with the store into memory with a plurality of physical page addresses indicating from what pages in a memory information was stored into a cache;

generating a self modifying code hit signal indicating a possibility of cache incoherency; and invalidating the information stored into the cache upon generation of the self modifying code hit signal.

24. The method of claim 23 wherein the plurality of physical page addresses are stored within an instruction translation lookaside buffer.

25. The method of claim 23 further comprising:

invalidating the information stored into an instruction pipeline from the cache upon generation of the self modifying code hit signal.

26. The method of claim 23 further comprising:

fetching instructions from memory to rewrite the information into the cache to obtain cache coherency.

27. A microprocessor including:

a memory controller to trigger a snoop if a store into a memory occurs; and

a translation lookaside buffer (TLB) having a content addressable memory (CAM) and receiving a physical address indicating the location where the store occurred in the memory, the TLB to store page translations between a linear page address and a physical page address pointing to a page of the memory having contents stored within a cache, the TLB including a CAM input port responsive to the snoop to compare the physical address received by the TLB with the physical page address stored therein.

28. The microprocessor of claim 27 wherein if it is determined that the physical address received by the TLB matches a physical page address stored within the TLB, indicating that information was modified within the memory correlating to information potentially located within the cache or a pipeline, and the microprocessor provides inclusion for the cache and the pipeline such that information within the cache and the pipeline are invalidated.

29. The microprocessor of claim 27 wherein the TLB includes data bits providing an indication as to what portion of a page in memory has its contents stored in the cache, and the microprocessor further includes snoop logic coupled to the TLB, the snoop logic to provide further comparison of the data bits with the physical address received by the TLB after a store into memory, the comparison provided by the snoop logic indicating if information was modified within a portion

of a page of memory correlating to information potentially located within the cache.

30. The microprocessor of claim 27 wherein the cache is a trace instruction cache and the information stored therein is instructions and the TLB is an instruction translation lookaside buffer (ITLB).